softnet news

information från SOFTNET USER GROUP nr 1/84 sid.1

adress: SUG, c/o Per Lundgren, ISY, Linköpings Universitet, 581 83 Linköping

REDAKTÖRENS RUTA

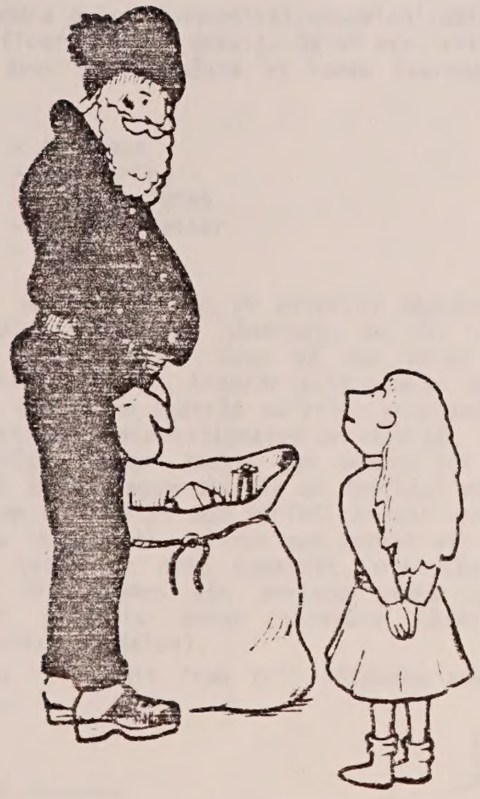
Nu är det inte så långt tills dess att SOFTNET går i luften. Radion håller på att få sin slutliga utformning, och ska inom kort sluttestas och caddas. Den och övrig SOFTNET-hårdvara finns beskriven under rubriken Hardware Corner av Ingvar Söderman, SM0LRF och Frithjof Qvigstad.

Framställning av kretskort till SOFTNET-noden kommer igång inom kort. Dessa går fr.o.m nu att beställa genom SUG.

SOFTNET NEWS står som synes på "egna ben" denna gång, men i fortsättningen finns vi som vanligt i AMSAT/SM:s informationsblad. Mer om det på annan plats i detta nr.

Till sist:
Vi syns väl på EXPRAN 84 ?!

God Fortsättning
önskar
red.



Packet-switching på gång?

News Release från American Radio Relay League

Call for Papers
Third Amateur Radio Computer Networking Conference

Konferensen arrangeras av ARRL i Trenton, N.J. den 15 april i samband med Trenton Computer Festival.

Manuskript färdiga för tryckning ska vara arrangörerna tillhanda senast den 1 mars. Den som avser att bidraga med ett konferensföredrag ska omgående rekvirera author's guide och samtidigt ange föredragets titel.

Adressen är
Paul L. Rinaldo, W4RI
American Radio Relay League
225 Main Street
NEWINGTON CT 06111

tel (203) 666-1541

Ämnesområden: Mailsystem, digitalt tal, packet radio, grafisk informationsöverföring m.m. med avseende på alla former av digital amatörradiokommunikation.

För ytterligare upplysningar kontakta Paul L. Rinaldo, W4RI.

****

Studentlitteratur ger till våren (1984) ut en inledande lärobok (på svenska) i FORTH-programmering med Paul Galli som författare.

****

SUMMARY

Hardware corner

The computer, modem and radio hardware of the Linköping University SOFTNET node implementation is outlined. Boards will be available in the spring of 1984.

MFORTH corner

A simple decompiler for the MFORTH system is described.

Deadlock corner

Routing strategies in distributed networks with limited amounts of topology information available to the nodes is considered. The simple "shortest distance" algorithm, and the "ZIP-code" algorithm are discussed.

HARDWARE CORNER

AV. INGVAR SÖDERMAN SMOLRF
FRITHJOF QVIGSTAD

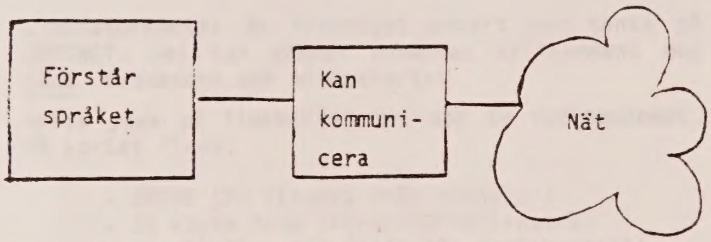
SOFTNET HÅRDVARA

Filosofiskt

Att ett så "mjukt" och flexibelt system som SOFTNET kräver en mjukvara som är både kraftfull, begriplig (även för användare) och flexibel (så att användare kan forma den efter behov) är självklart. Men även på hårdvaran ställs liknande krav. Det vi vill ha är en apparat som kan modifieras (programmeras) till önskad funktion. Vad är då önskad funktion? Ja, SOFTNET är faktiskt så mjukt (luddigt?), att det enda som är spikat är språket med vilket noderna kommunicerar. Nu är det ju så, att det krävs mer än bara ett gemensamt språk för att kommunicera; exempelvis hårdvara.

Realistiskt

Låt oss dela upp noden i två delar. Den ena delen har endast kravet att den förstår språket och kan utföra instruktioner givna i detta språk. Här kan vi använda en hårdvara som kan (programmeras att) förstå språket.

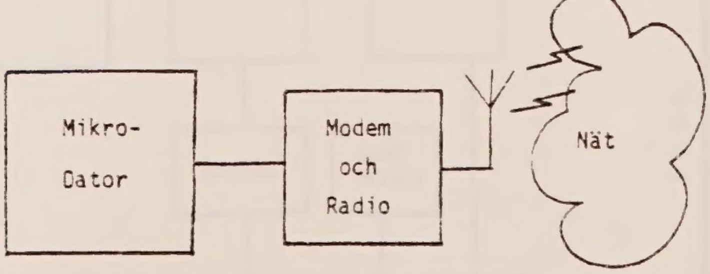


Den andra delen (kommunikationsdelen) däremot måste specificeras mer i detalj. Om vi exv. vill kommunicera över radio, måste vi komma överens om saker som:

- frekvens
- modulering
- datahastighet
- sändareffekter
- m.m.

Dessa parametrar har vi givetvis bestämt, och de behandlas av Ingvar Söderman. Nu får vi förstås vissa prestandakrav även på den delen som bara "förstår språket". Framför allt kommer datahastigheten att ge en ledtråd om vilka prestanda man bör eftersträva. Datahastigheten är vald så, att en nod får "några tusen" tecken per sekund att behandla, vilket är en lagom dos för en "hyfsad" mikrodator. Även om huvudmålet med SOFTNET är att använda ovan nämnda radiokanal, så kan man mycket väl tänka sig andra typer av nät; kabelnät, mikrovågsnät m.m. Dessa nät kommer att använda samma språk, men kräver givetvis annan hårdvara (åtminstone i kommunikationsdelen).

Vi har nu kommit fram till följande nodimplementation.



Här kommer jag endast att beskriva datordelen, modemet och radion behandlas av Ingvar Söderman.

Praktiskt

Vi har nu vissa krav, som vår (mikrodator-) implementation skall uppfylla; förstå språket, klara datahastigheten utan att bli flaskhals (under normala förhållanden) och gärna ett lågt pris.

Låt oss se mer i detalj på noddatorns uppgifter.

- Ta emot om inkommande paket utan fördröjning
- Sända paket (när kanalen är ledig)
- Tolka paketens innehåll och utföra instruktionerna.

De två första uppgifterna är starkt knutna till varandra och i princip oberoende av den tredje uppgiften.

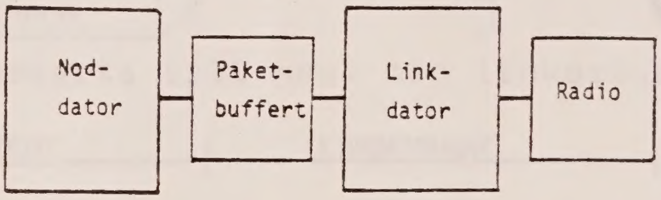
Dessa oberoende uppgifter kan ses som två olika processer i noden. Den ena (sändning och mottagning) kallar vi för LINKPROCESSEN, och den andra kallas NODPROCESSEN.

Man inser att linkprocessen måste vara ständigt beredd på att ett paket kan komma, eller att det skall sändas ett paket. Den har alltså hårda realtidskrav.

Nodprocessen däremot har inga kritiska realtidskrav, men bör vara "snabb" för att ge korta svarstider.

Elektriskt

Den SOFTNET-nod vi har byggt följer mycket väl ovanstående uppdelning. Datordelen är implementerad med två oberoende mikrodatorer: Linkdatorn och noddatorn, som båda två kan läsa och skriva i en paketbuffert.



För närvarande är systemet uppbyggt på tre kretskort (nod, link, paketbuffert) som är 100x220 mm.

Noddatorkortet (CC09)

Noddatorn är en enkortsdator som är framtagen speciellt för programutveckling i FORTH. På kortet finns:

- Motorala 6809 CPU
- 8 kByte Prom med M-Forth
- 4 kByte ledigt Prom
- 8 kByte statiskt RAM
- terminalport (ACIA)
- parallellportar (VIA)
- bandspelaranslutning
- watchdog
- bussanslutning för utbyggnad.

Detta är alltså ett generellt datorkort, ej speciellt utvecklat för SOFTNET.

softnet news
information från SOFTNET USER GROUP

HARDWARE CORNER

forts.

Paketbuffert

Paketbuffertkortet är ett minneskort som kan användas av två datorer samtidigt, utan någon som helst fördröjning. Detta fungerar genom att minneskapslarna är dubbelt så snabba som CPU-erna. På kortet finns:

- 8 x 2 kByte statiskt RAM
- 2 x 8 kByte statiskt RAM
- Bussanslutning mot noddatorbussen
- Bussanslutning mot linkdatorbussen
- 2 x 4 kByte Prom åtkomligt endast från noddatorn.
- Ackumulator för minnesbackup.

Var och en av de tio RAM'en kan vara "backupade" och/eller skrivskyddade. Dessa funktioner väljs med byglar vid varje minneskapsel.

Detta kort går naturligtvis utmärkt att använda som ett extraminne till CC09, vilket ger en Forth-dator med 40 kByte RAM, 20 kByte Prom o.s.v. på två kort.

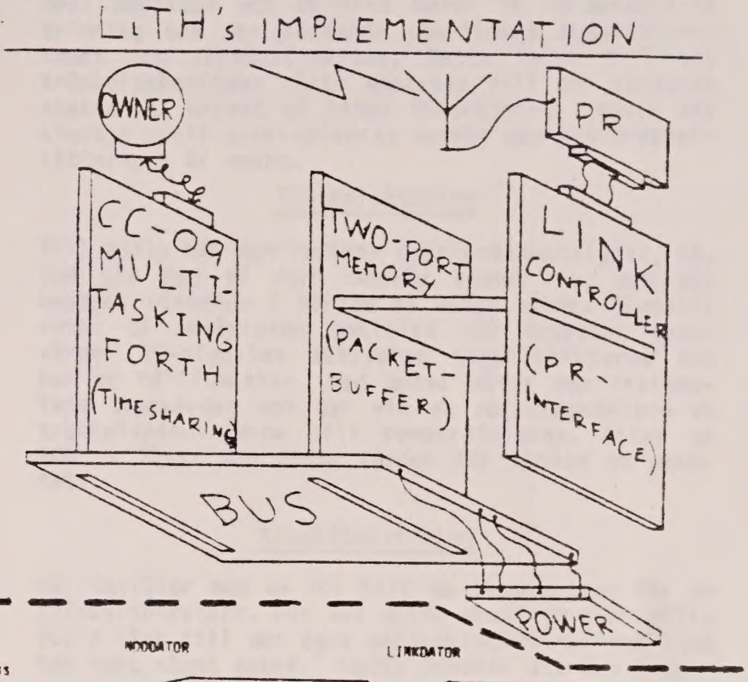
Linkdator

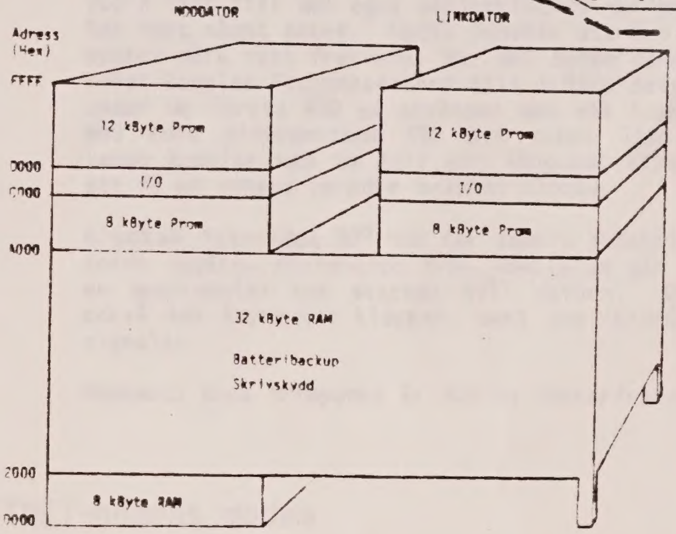
Linkdatorkortet är framtaget enbart med tanke på SOFTNET. Det kan endast användas tillsammans med både noddatorn och minneskortet.

Halva ytan på linkkortet tas upp av radiomodemet. På kortet finns:

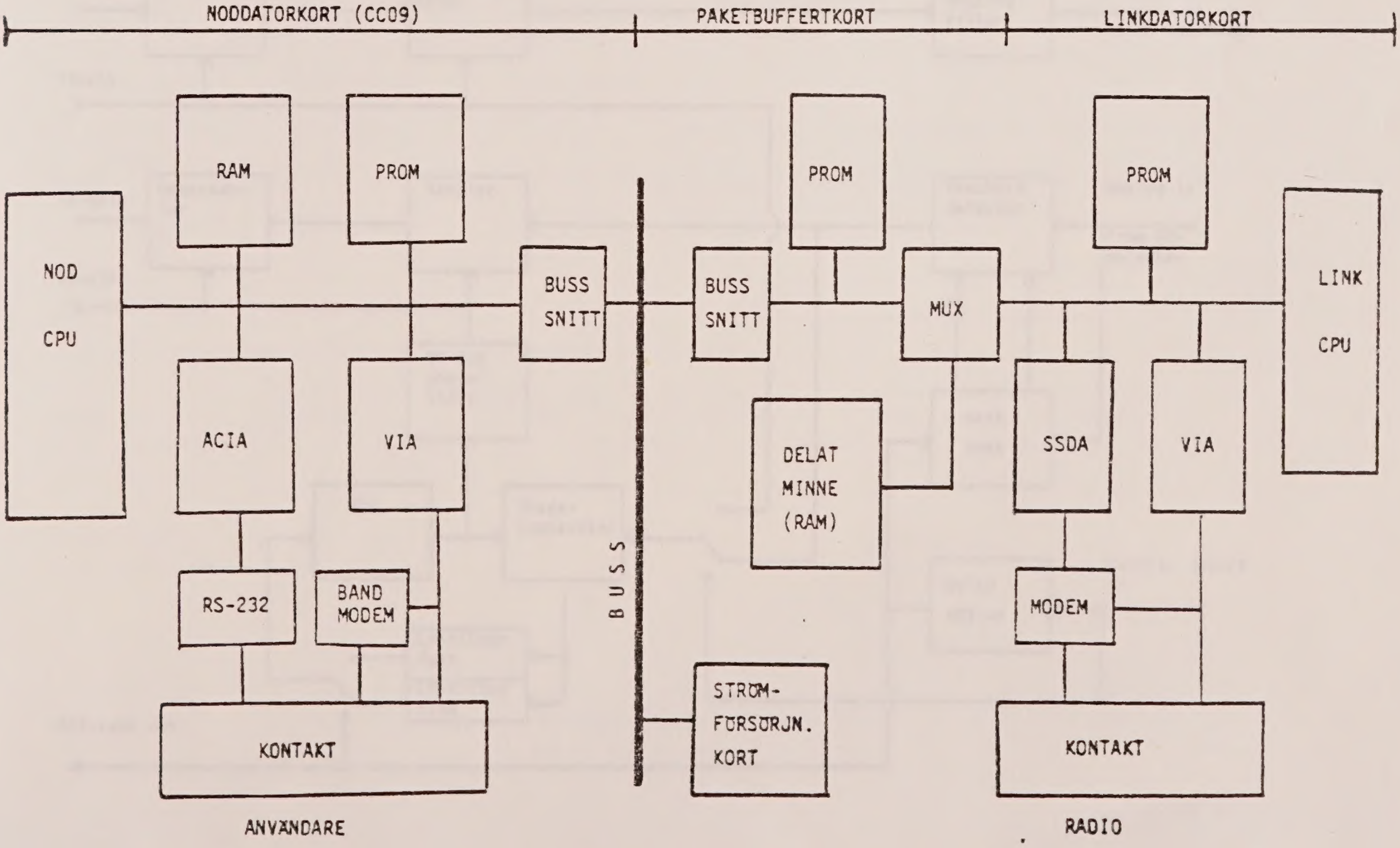
- 6809E CPU (klocka från noddator)
- 20 kByte Prom (Forth+SOFTNET+extra)
- parallellportar (VIA) för radiokontroll
- serieport (SSDA) till modemet
- modem

Linkdatorkortet ansluts med en kontakt direkt till minneskortet.





Minneskarta till nod- och linkdator



HARDWARE CORNER

forts.

MODEMBESKRIVNING

Målsättningen med modemkonstruktionen var att åstadkomma ett modem med följande, något svårförenliga egenskaper.

Datahastighet 100 kbit/s	
Synkronöverföring	
Liten fördröjning	Kort T/R switchtid
Bandbreddssnålt	
Robust	Låg komponentkänslighet
Lättbyggt	Få trimpunkter
Billigt	

Sändardelen

För att kombinera datahastighets- och bandbreddskraven valdes duobinärkodning, vilket innebär att den binära signalen kodas om till 3 nivåer och dubbelt så långa pulser. Dessa pulser rundas sedan av i ett lågpassfilter. Detta ger en halvering av bandbredden. jämfört med binär signalering, och endast en liten ökning av komplexiteten.

Scramblern som föregår duobinärkodaren kodar om successiva "nollor" till en 1,0,1,0,1,... följd. Detta för att mottagaren skall kunna bibehålla synkronism lättare, då följder av "nollor" har hög sannolikhet.

Mottagardelen

Denna är betydligt mer komplicerad än sändardelen, här skall klockan återvinnas och signalen som kommer hit har fått tillskott av brus och störningar. Dess amplitud och DC-nivå beror av FM-detektorns trimning och den sändande stationens modulationsindex och bärvågsfrekvens. Detta leder till att tröskelspänningen måste anpassas till den sändande stationen. Kravet på liten fördröjning innebär att klockan skall synkroniseras snabbt och att tröskelsättningen är snabb.

Tröskelsättning

Till hjälp har man radions carrierdetectsignal, CD, som går hög så fort bärvåg kommer in, och ett bestämt utseende i början på varje paket. Signalen antar då omväxlande positivt och negativt toppvärde. CD-signalen aktiverar topplikriktarna som har en hållfunktion. Med dessa mäter man insignalens toppvärden och tar via en spänningsdelare ut tröskelspänningarna till komparatorerna. Efter ca 400 µs låser man dessa värden för resten av paketet.

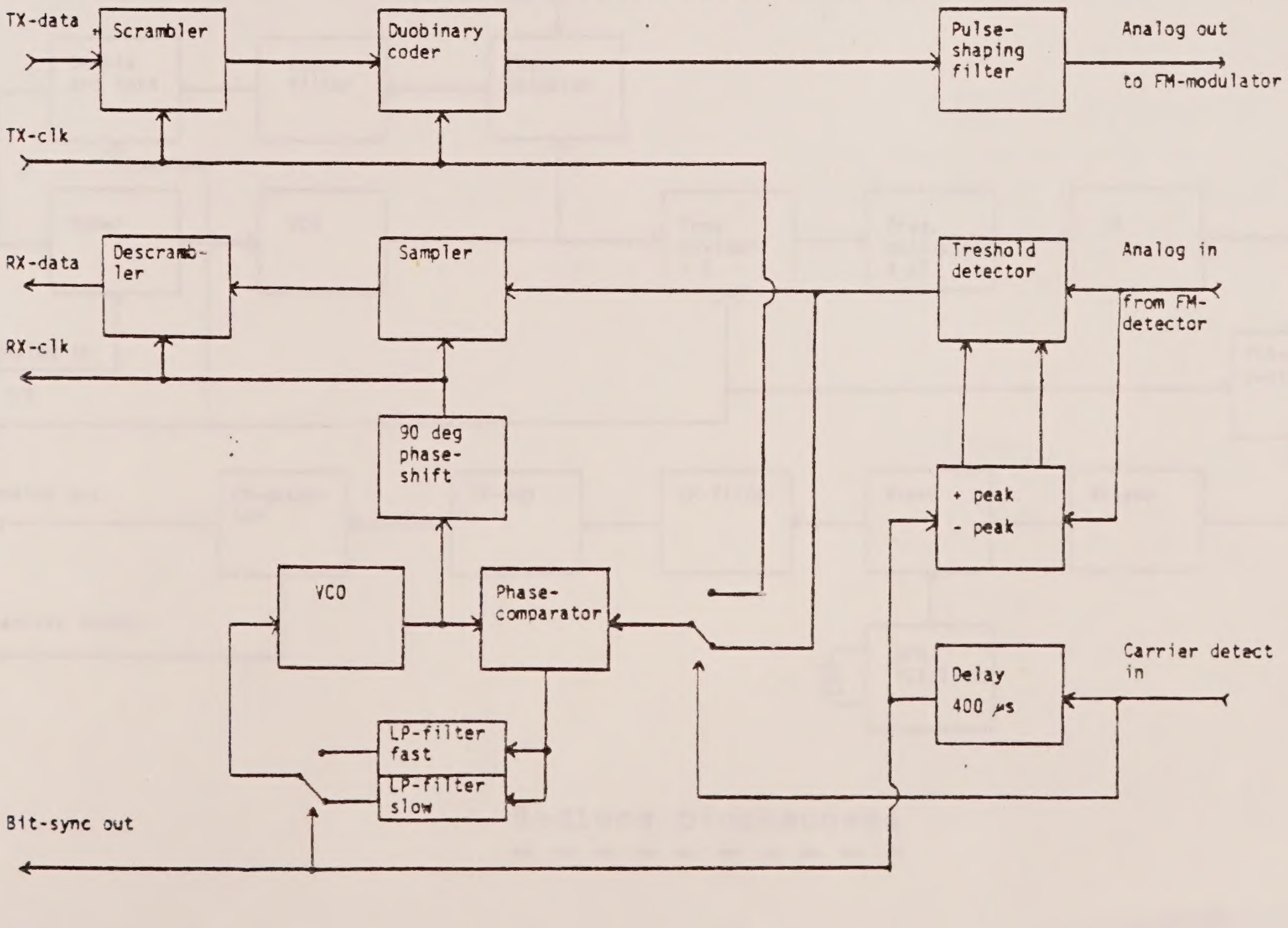
Klockåtervinning

Här faslåser man en VCO till de flanker man får ur tröskeldetektorn. För att detta skall gå fort hålls VCO'n låst till den egna sändarklockan när man inte tar emot något paket. Detta innebär att den ligger mycket nära rätt frekvens. När det sedan kommer ett paket kopplas faskomparatorn till tröskeldetektorn. Under de första 400 µs använder man ett loopfilter med kort tidskonstant för att kunna låsa fort, sedan kopplar man om till ett långsamt filter för att få en renare (mindre brusig) klocka.

Klockan fasvrides 90° och får sampla tröskeldetektorns utgång. Utsignalen från samplaren går igenom en descrambler och skickas till datorn. Dit går också den återvunna klockan, samt den fördröjda CD signalen.

Modemets enda trimpunkt är VCO'ns centerfrekvens.

Blockschema över SOFTNET-nodens modem



HARDWARE CORNER

forts.

Radion

Radion är så när som på modulatorn och MF-filtret helt konventionell.

Vid smalbands FM brukar man ju dra en kristalloscillator med en varicap, men det går inte här. Vi har ju ca 60 kHz bandbredd på insignalen. Istället använder vi en PLL-koppling med sample and hold. När sändningen startar kopplas holdfunktionen in och datasignalen adderas till VCO'ns styrspänning. När man inte sänder är återkopplingsslingan sluten. VCO'n och referensoscillatorerna går hela tiden och sändningen slås till och från genom att styra delaren. Detta för att få snabb start. Att starta en kristalloscillator tar ca 300 us. Detta förfarande tar < 1 us.

Genom att dela med ett jämnt tal och multiplicera med ett udda slipper man få övertonen från oscillatorn in i mottagaren.

Mottagaren är en vanlig enkelsuper med ett brett, ca 250 kHz faslinjärt MF-filter. Detta är nödvändigt för att inte distordera den bredbandiga datasignalen.

För antennomkopplingen används PIN-dioder, vilka har mycket snabb omslagstid.

*

: M-FORTH CORNER ;

AV JENS ZANDER

Till denna spalt utlovades en enkel men effektiv dekompilator. En dekompilator är invers-operationen till FORTH systemets :-kompilator. Med hjälp av : definierar vi ju nya ord som en lista av tidigare definierade ord. Dessa redan befintliga ord läggs inte ut i det nyskapade ordet som de är, utan man lägger endast ut adresser till dessa. Ett enkelt exempel:

```
:     SQR DUP * . ;
```

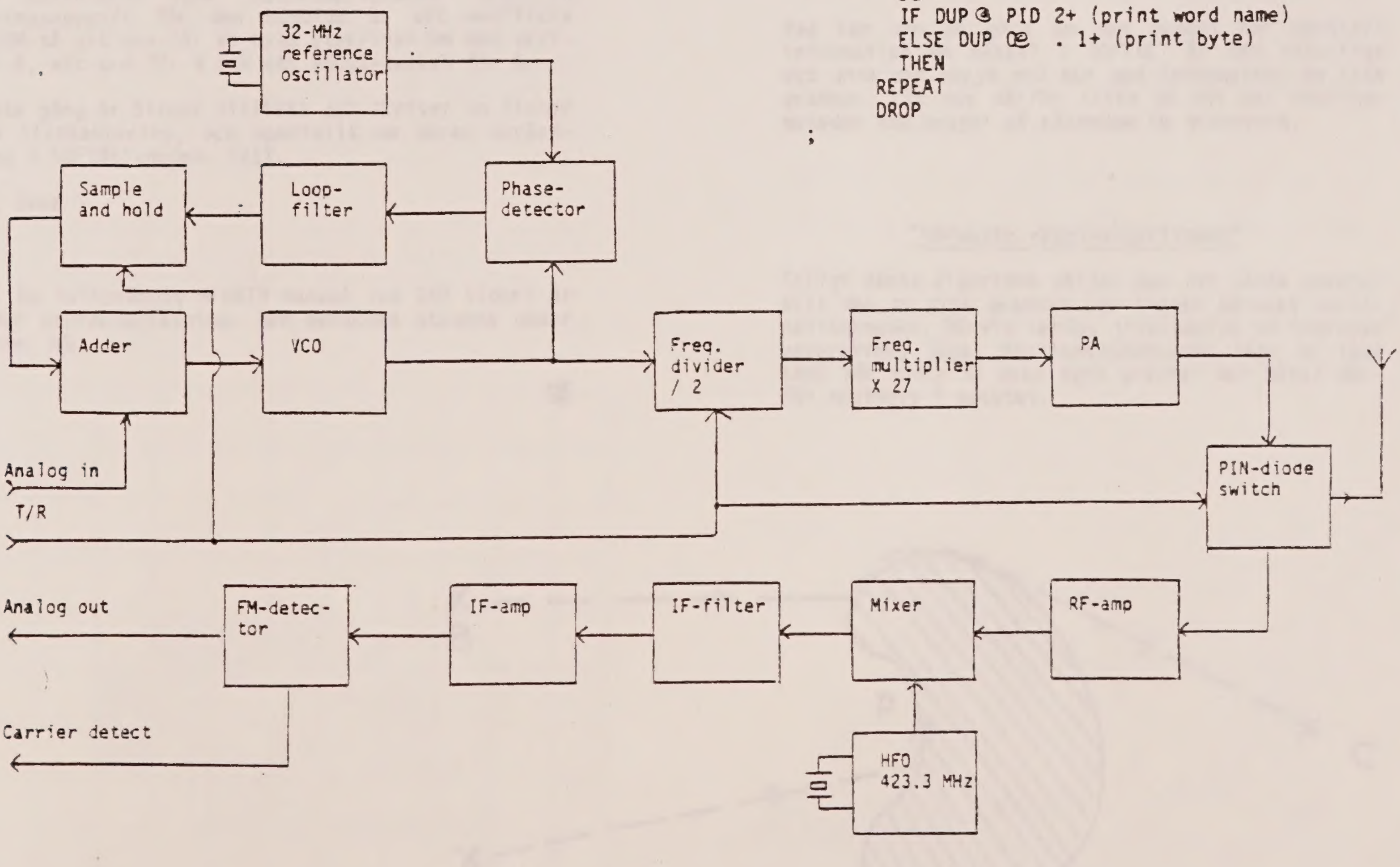
83	SQR	•	•	<DUP>	<*>	<.>	<;S>

Vi ser i exemplet hur addresserna till DUP, * och läggs ut i den nya definitionen. Notera att ;S kompileras in av ; för att avsluta order.

Dekompilern gör alltså det omvända, den tar de kryptiska addresserna och skriver ut namnen på de ord de motsvarar. Nedan återges koden till en enkel dekompiler som den återfinns i standard options PROMet till MFORTH 3.5

```
      : PID 2+ NFA ID. ;
      : ?CODE DUP 2- @ = 33 ?ERROR ;

: DECOM
      [COMPILE] '          (get PFA)
      DUP ?CODE            (check if codeword)
      BEGIN
        KEY                (wait for keyboard input)
        DUP 13 = NOT       (check if CR )
      WHILE
        BL =
        IF DUP @ PID 2+ (print word name)
        ELSE DUP C@  . 1+ (print byte)
        THEN
      REPEAT
      DROP
;
```



Radions blockschema

Digitized by the Internet Archive
in 2025 with funding from
Amateur Radio Digital Communications, Grant 151

https://archive.org/details/softnet-news-1984-01

: M-FORTH CORNER ;

forts.

PID tar en kompilerad adress från stacken och skriver ut namnet på det motsvarande ordet. ?CODE kollar om ordet är ett maskinkodsord, i vilket fall dekompileringen avbryts. DECOM används nu på följande sätt. Låt oss t.ex. dekompilera ordet SQR som vi definierade här ovan. Vi skriver

DECOM SQR (return)

Om vi nu skriver ett mellanslag, så skrivs nästa kompilerade ord ut. Fortsätt att skriva mellanslag tills ;S dyker upp. Här, och vid alla andra tillfällen kan vi avbryta genom att skriva return.

DECOM SQR DUP * . ;S

Om vi skriver något annat än blank eller return kommer nästa byte i minnet att skrivas ut. Detta är bra att ha, om man kommer till ett ord som följs av inkompilerade data. Genom att skriva ut dessa data kan man passera dem, en byte i taget. Ett exempel är följande:

: TEST 10 0 DO I. LOOP;

DECOM TEST LIT 0 10 0 (DO) I . (LOOP) 255 250 ; S

Understrukna siffror har tagits fram byte för byte. (Notera hur 0:an lagrats!). På detta sätt kan man interaktivt dekompilera sina ord. "Fasta" dekompilers som känner igen LIT, (LOOP) m.m. fungerar automatiskt, men råkar i svårigheter om man definierar egna "datakompilerande" ord. En övningsuppgift för den händige är att modifiera DECOM så att man får en byte utskriven om man skriver B, ett ord för W och ett ASCII-tecken för C.

Nästa gång är Birger tillbaka och skriver om listor och listhantering, och speciellt om deras användning i SOFTNET-noden. EXIT.

73, Jens

PS. En fullständig MFORTH manual (ca 150 sidor) är under korrekturläsning. Den beräknas utkomma under våren. DS

*

DEADLOCK CORNER

av Robert Forchheimer

THE DEADLOCK CORNER

I förra numret av SOFTNET NEWS beskrev jag hur ett system av FORTH-noder kan fås att överföra paket mellan sig. Jag nämnde också den "magiska" funktionen G% som ger bästa nod att vidarebefordra ett paket till, då det skall gå en längre sträcka i nätet.

Problemet att finna en lämplig sådan funktion (routingproblemet) är inte trivialt. I allmänhet har en nod som skall ta ett beslut om bästa granne endast begränsad information om hur nätet ser ut, vilka noder som för närvarande är aktiva, trafikbelastning etc. Vi kan betrakta följande tre olika fall:

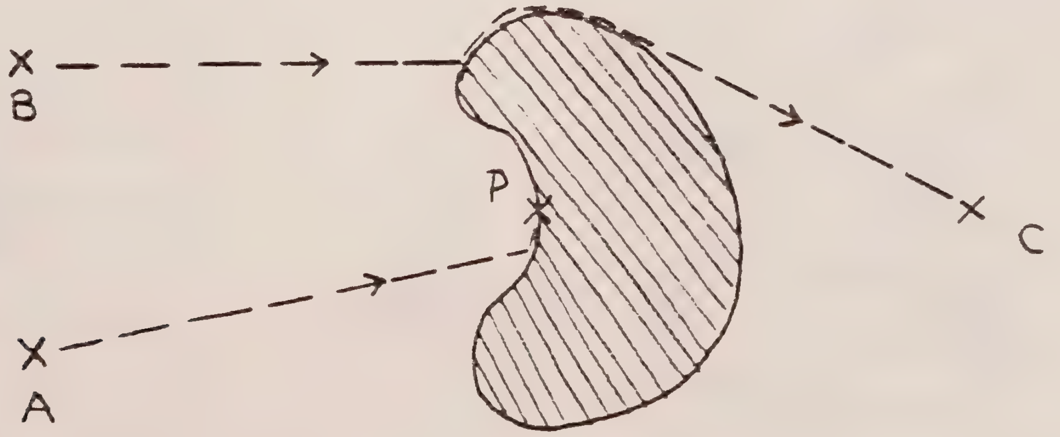
1. Noden har fullständig information om nätet.
2. Noden har partiell information om nätet.
3. Noden har ingen information om nätet.

Det är uppenbart att i fall 1 så ges goda möjligheter att räkna ut den optimala vägen. Men är det möjligt att i alla noder ha fullständig information om nätet? Knappast! I ett statiskt nät med givna noder så kan nätets topologi mycket väl vara känd men den momentana trafikbelastningen är fortfarande okänd. Denna information kan heller inte överföras i nätet utan att påverka sig själv. Fall 1 är således ett idealtillstånd vilket vi endast kan närma oss genom att utnyttja en allt större del av nätets kapacitet till att förmedla information om nätet självt.

Vad kan man då göra om man endast har partiell information om nätet? I SOFTNET är det naturligt att anta att varje nod har god information om sina grannar. Låt oss därför titta på ett par routingmetoder som bygger på kännedom om grannarna.

"Närmaste vägen-algorithmen"

Enligt denna algorithm väljer man att sända paketet till den av sina grannar som ligger närmast destinationsnoden. Därvid behövs information om nodernas geografiska läge. Destinationsnodens läge är inte känd för andra än dess egna grannar och måste därför medföras i paketet.



DEADLOCK CORNER

forts.

Figuren visar vad som händer i två olika fall. Noderna A,B sänder paket till C. Nätet förutsättes vara mycket finmaskigt och kan makroskopiskt betraktas som en kontinuerlig yta. Dock saknas noder i det streckade området (hundkapplöpningsbana).

Paket från B tar sig förbi hindret och når destinationen C. Paket från A däremot kommer att "fastna" i nod P. Detta beror på att denna nod inte längre kan vidarebefordra paketet då samtliga grannar ligger längre från C än vad P själv gör.

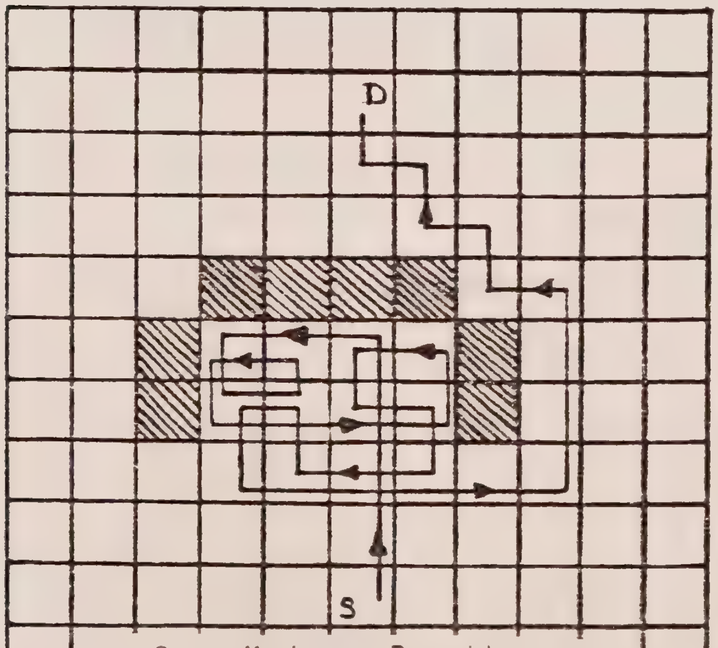
Det är således uppenbart att denna enkla routing-algorithm kan råka ut för svårigheter. Hur skulle då en bättre metod se ut?

I skriften "Design Consideration of a distributed Packet Radio Network" beskriver jag några tänkbara metoder. Principen går ut på att lämna nod P ("the deadlock corner") längst randen till det streckade området även om detta medför att paketet tillfälligt fjärmas från C. Ett sådant byte av strategi är möjligt eftersom nod P inser att ett felaktigt tillstånd har uppstått. Det återstår dock att signalera till de andra noderna att "randföljningsstrategi" skall användas i stället för "närmaste vägen"-strategi. En sådan signalering göres lämpligen genom att sätta en flagga i paketet.

En annorlunda metod för att finna en väg ut ur konkaviteter av ovanstående slag har föreslagits av Sven Engnell. Metoden bygger på att paketet, efter att ha hamnat i P, letar sig ut igen enligt en trädsökningsalgorithm. Varje nod som har vidarebefordrat paketet lagrar temporär information om att paketet har passerat. Om samma paket skulle komma tillbaka så väljer noden att sända paketet till en annan granne än den som tidigare användes. Effekten blir att paketet systematiskt kommer att gå till olika noder ända tills hindret har passerats.

Nedanstående bild som är hämtad från Svens rapport "Förslag till routingmetoder i packet-switching nät" visar vad som händer i ett simulerat fall.

För enkelhets skull antas noderna ligga i ett kvadratiskt gitter.



S=avsändare D=mottagare

Global routing-information

Det är uppenbart att routing baserad enbart på kunskap om grannarna blir en smula ointelligent. Detta beror naturligtvis på att varje nod endast har lokal kunskap om nätet. Bättre resultat kan förväntas om noderna har tillgång till viss global information, såsom exempelvis förekomsten och utseendet av hinder, nätets topologi etc. Hur man skaffar sig sådan information och på vilket sätt den kan användas för routing tänker jag återkomma till i nästa nummer. Medan Du väntar på detta så fundera gärna på hur en routingalgorithm skulle kunna se ut för fallet (3) som jag inte har gått in på. Här gäller alltså att noderna inte har någon som helst information om nätet i övrigt!

Kommer Du på någon bra lösning så skicka den till mig på adress: Institutionen för Systemteknik, LiTH, 581 83 Linköping. Jag tar även tacksamt emot idéer, uppslag och övriga kommentarer som passar in i profilen för denna spalt.

Med bästa Julhälsning

Robert

****

SOFTNET NEWS,

informationsorganet för SOFTNET-intresserade, kommer i fortsättningen att vara en del av AMSAT-info som ges ut av AMSAT/SM.

Detta har sin grund i att SUG har bildats som en projektgrupp inom denna förening.

AMSAT/SM är en förening för radioamatörer som bl.a sysslar med avancerad radioteknik och satellitkommunikation.

Medlemsavgiften i AMSAT/SM är 50 kr och betalas på pg 83 37 78 - 4. Märk talongen "SOFTNET USER GROUP".

EXPERIMENTAL PACKET RADIO NETWORKS
INTERNATIONAL SYMPOSIUM, LINKÖPING, SWEDEN
With a Separate Workshop On SOFTNET

Call For Papers

The symposium will be held at Linköping University, Linköping, Sweden - May 26-27, 1984

Authors...

You are invited to submit an original paper for consideration at the Experimental Packet Radio Networks International Symposium. Your paper should address a suggested area of interest.

SCHEDULE

Notice of intent, due .... Jan. 31

Reception of authors kit, due .... Febr. 29

Camera ready manuscripts, due .... March 31

AREAS OF INTEREST

1. *Experimental Packet Radio*
2. *Network routing and control*
3. *Distributed Processing*
4. *Channel access protocols*
5. *Standard proposals*
6. *SOFTNET*

INSTRUCTIONS

1. *Complete and mail the Notice of Intent below.*
2. *Send manuscripts to*

SOFTNET User Group
Dept. of Electrical Engineering
Linköping University
S-581 83 Linköping, Sweden

For additional information dial

Nat. 013-28 10 00

Int. + 46 13 28 10 00

and ask for
Per Lundgren or
Gunilla Svahn (secretary)
Dept. of E.E.

Notice Of Intent

Please complete this form and mail it to

SOFTNET User Group
Dept. of Electrical Engineering
Linköping University
S-581 83 Linköping, Sweden

☐ *I'am planning to attend the symposium*

☐ *I'am planning to submit a paper on*

....

....

....

....

Name ....

Address ....

....

Telephone ....

KÖP KRETSKORT TILL SOFTNET-NODEN

Snart startar framställningen av kretskort till SOFTNET-noden! Gör en beställning nu, så är du säker på att komma med i första omgången.
Först en kort instruktion:

Del 1: Beställning av kretskort. Beställningen betraktas som giltig endast om Du skickar in halva beloppet på de beställda korten till SUG.

Del 2: Avsikten med den här delen är att undersöka hur många som vill köpa färdigbyggda kort m.m. Framförallt gäller detta kanske radion, som ju kräver rätt omfattande trimning.

Produktionen av noddator-, link- och minneskorten kommer igång i mitten av februari. Radiokortet beräknas kunna börja tillverkas i slutet av samma månad.

Del 1.

Kretskortsbeställning.
Fyll i vilka kort Du vill köpa. Skicka in den här lappen till SUG senast den 26/1. För att vi ska slippa ligga ute med för mycket pengar ska Du också sätta in HALVA BELOPPET på de beställda korten på pg 18 34 15 - 9 Linköpings Universitet.

VIKTIGT! Märk pg-talongen tydligt "SOFTNET", samt skriv ditt namn och din adress.

Antal		Pris/kort (c:a)	Du betalar i förskott/kort
_____	Nodkort(CC-09)	160 kr	80 kr
_____	Minneskort	160 kr	80 kr
_____	Link-modemkort	160 kr	80 kr
_____	Radiokort	160 kr	80 kr

Del 2.

Är Du intresserad av att köpa komponentsatser och/eller färdigbyggda kort ? Fyll i Dina önskemål.
OBS!! Denna service tillhandahålles endast i mån av tillräckligt intresse.

	Komponent-sats	Färdigbyggt och trimmat
Link-modem	!__!	!__!
Minne	!__!	!__!
Noddator	!__!	!__!
Radio	!__!	!__!

Namn ______________________________

Adress ______________________________

____

Skickas före 26/1 till:
SUG
c/o Per Lundgren
Inst. för systemteknik
Linköpings Universitet
581 83 Linköping

[illegible]

Del 1. [illegible]

Del 2. [illegible]

[illegible]

Del 1.

[illegible]

[illegible]

Antal	[illegible]	[illegible]
___ [illegible]	150 kr	30 kr
___ [illegible]	150 kr	30 kr
___ [illegible]	150 kr	30 kr
___ [illegible]	150 kr	30 kr

Del 2.

[illegible]

	[illegible]	[illegible]
[illegible]	☐	☐
[illegible]	☐	☐
[illegible]	☐	☐
[illegible]	☐	☐

Namn ______________________

Adress ______________________

____

[illegible]
c/o Per [illegible]
[illegible]
[illegible] Universitet
[illegible]